

Sistemas Distribuidos

Tarea 2: Informe Comparativo

Francisca Ramírez
Ricardo Carrasco

enero 2020

1. Introducción

A continuación se presentaran las principales características técnicas al implementar un Chat usando dos *frameworks* de micro servicios. El objetivo de este informe es presentar una recomendación sobre cual de las dos arquitecturas distribuidas es mejor para la implementación del chat.

Este documento comparará dos tipos de infraestructura de micro servicios, *Remote Procedures* y *Asynchronous Messaging*. La implementación llevada a cabo consiste en un chat entre varias personas con un servidor central encargado de coordinar los mensajes.

2. Remote Procedures - gRPC

gRPC es un *framework* usado para realizar llamadas a procedimientos remotos, es capaz de general enlaces multi plataformas entre los clientes y el servidor y dispone de una gran variedad de lenguajes. Para la creación del chat, el *framework* fue usado con el fin de conectar al cliente y el servidor usando traspaso de mensajes.

En la tarea RPC es usado por ejemplo al obtener la lista de mensajes, ya que al ser un sistema síncrono, el cliente queda a la espera hasta recibir la respuesta del servidor. El uso de este *framework* presenta pros y contras, a favor tiene que permite un sistema de mensajería instantánea, pero al usar métodos síncronos el sistema se vuelve más lento cuando la lista de mensajes y/o usuarios se volvía más larga.

3. Asynchronous Messaging - RabbitMQ

RabbitMQ permite conectar distintas aplicaciones sin conocer los detalles de arquitectura de cada una, esto lo logra con la validación, transformación y ruteo de mensajes. Para llevar a cabo lo recién mencionado RabbitMQ usa los siguientes elementos: Consumidor/Productor (rutean con un mensaje para la cola), Cola (Entregan el mensaje al consumidor) y Exchanges (transmiten mensajes a más de una cola). Usar RabbitMQ permite distribuir mensajes de forma más rápida, esto se convierte en una de las ventajas de esta implementación es que permite la distribución de mensajes de manera más eficiente mediante un broadcast y ruteos. La desventaja es que si bien, esto funciona de forma asíncrona habrán procesos que requieren ser síncronos y lograrlo se debe definir muchas restricciones.

4. Recomendación

En base a lo mencionado en los puntos anteriores, nuestra recomendación es usar gRPC para la realización del chat. Rabbit es un buen framework que facilita la distribución de mensajes entre usuarios, sin embargo no se adecua a lo que requiere un chat en tiempo real. Usando RPC se facilitan varias implementaciones de procesos para comunicación cliente-servidor, además gRPC permite modularizar de forma conveniente las implementaciones llevadas a cabo.